

NightHub Studio

Local AI workstation – feature overview

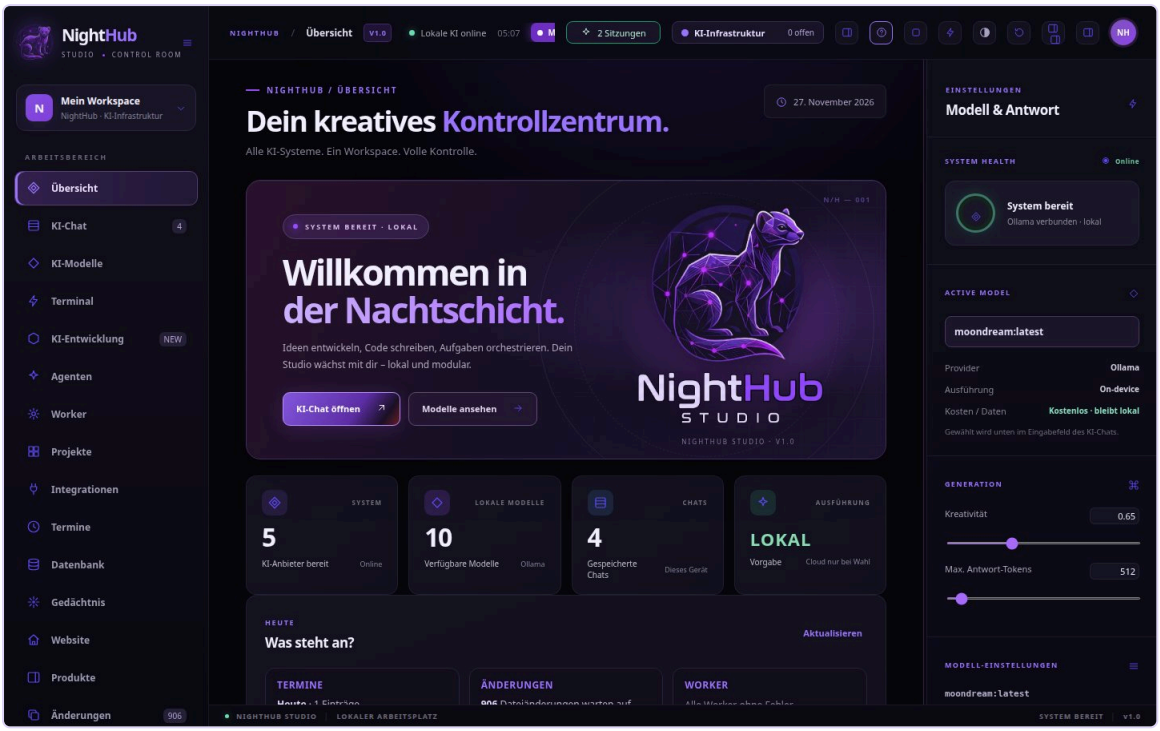
Desktop app (Tauri) with a local Python hub. Everything runs on my own machine; cloud AI only when set up and explicitly chosen. Every action an AI takes on programs or files needs approval and is logged.

Version 1.0, November 2026 · own project of NightHub Studio · no client data

Technology

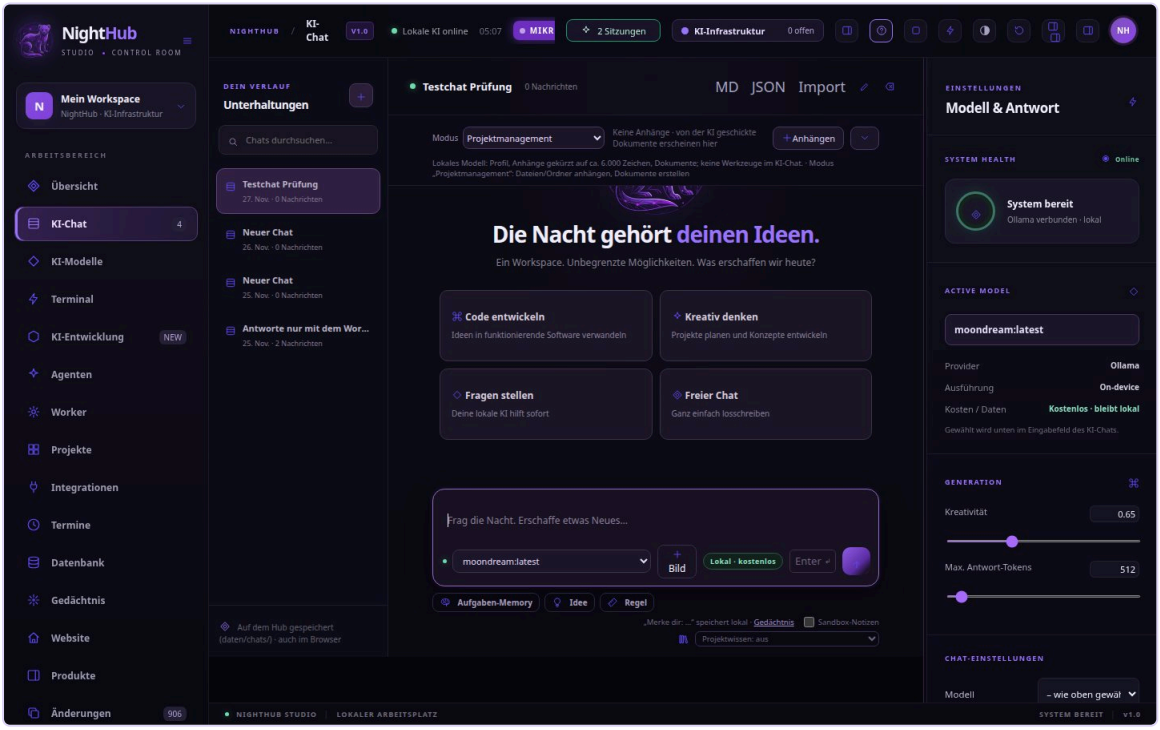
Interface	Tauri (Rust) desktop window, HTML/CSS/JavaScript, tray icon, single instance, split screen across two monitors
Hub	Python service on 127.0.0.1 (systemd user service), more than a hundred modules, each tested
Local AI	Ollama with small models on a 4 GB graphics card, Whisper (speech to text) and Piper (text to speech)
AI providers	Claude Code, Codex, GitHub Copilot, cloud APIs (OpenRouter, Groq, Mistral and others), Notion agents
For other AIs	own MCP server: ask roles, memory, appointments, code snippets, workers, integrations by permission
Data	SQLite (memory with full-text search, ID registry), Notion databases via API, JSON settings with backups

01 Overview



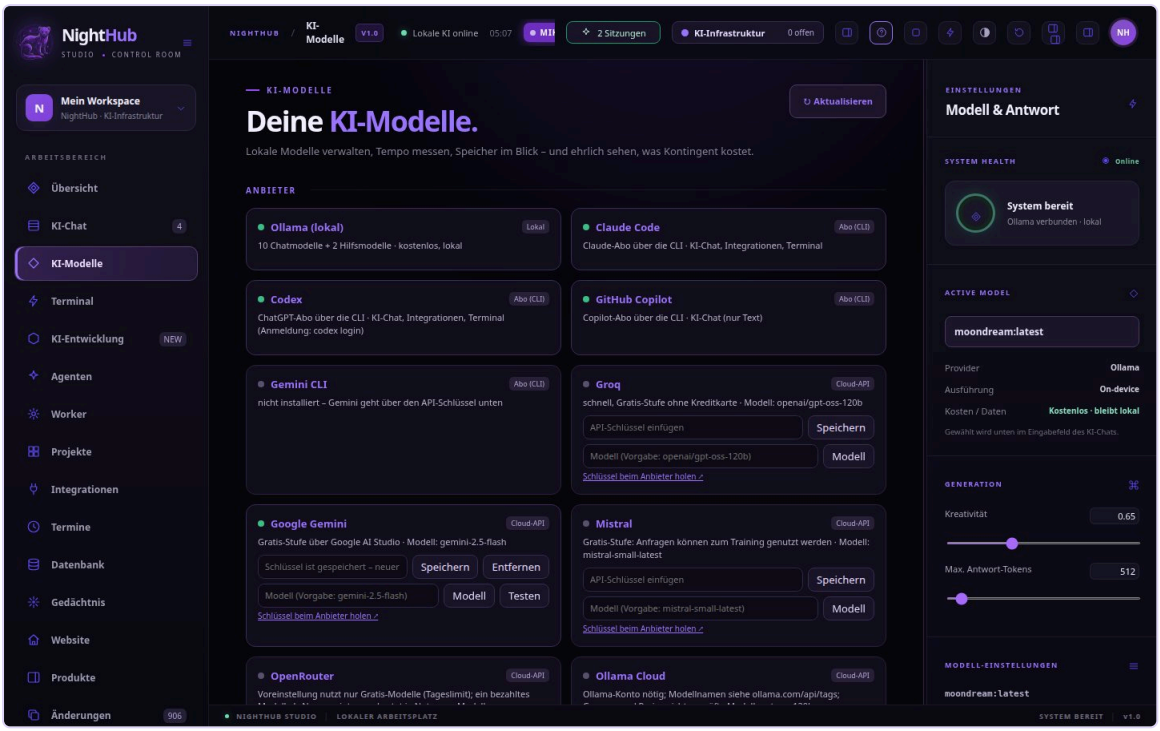
Start page with system status: connected AI providers, local models, chats and where things run. Below: what is due today – appointments, changes, workers. On the right of every view: model settings (creativity, answer length).

02 AI chat



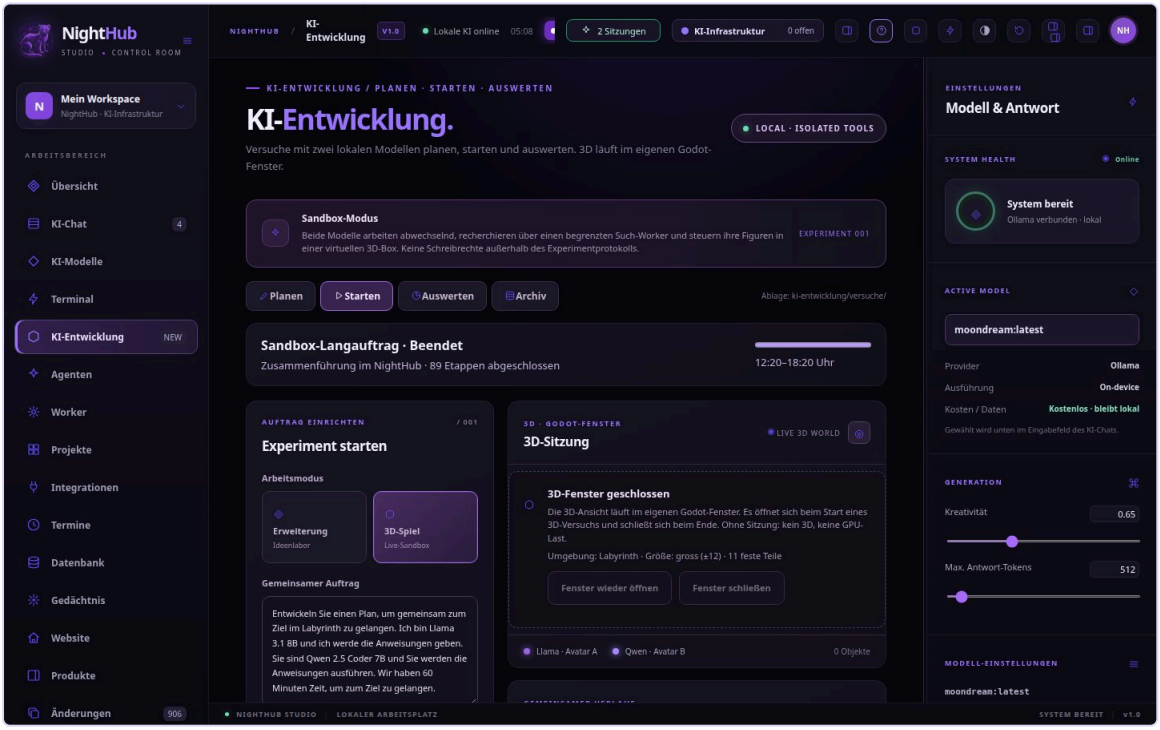
One chat for all models: local via Ollama, subscriptions (Claude Code, Codex, Copilot), cloud APIs and Notion agents. History with search, modes such as project management, attachments and images. A label always shows whether an answer stays local and free.

03 AI models



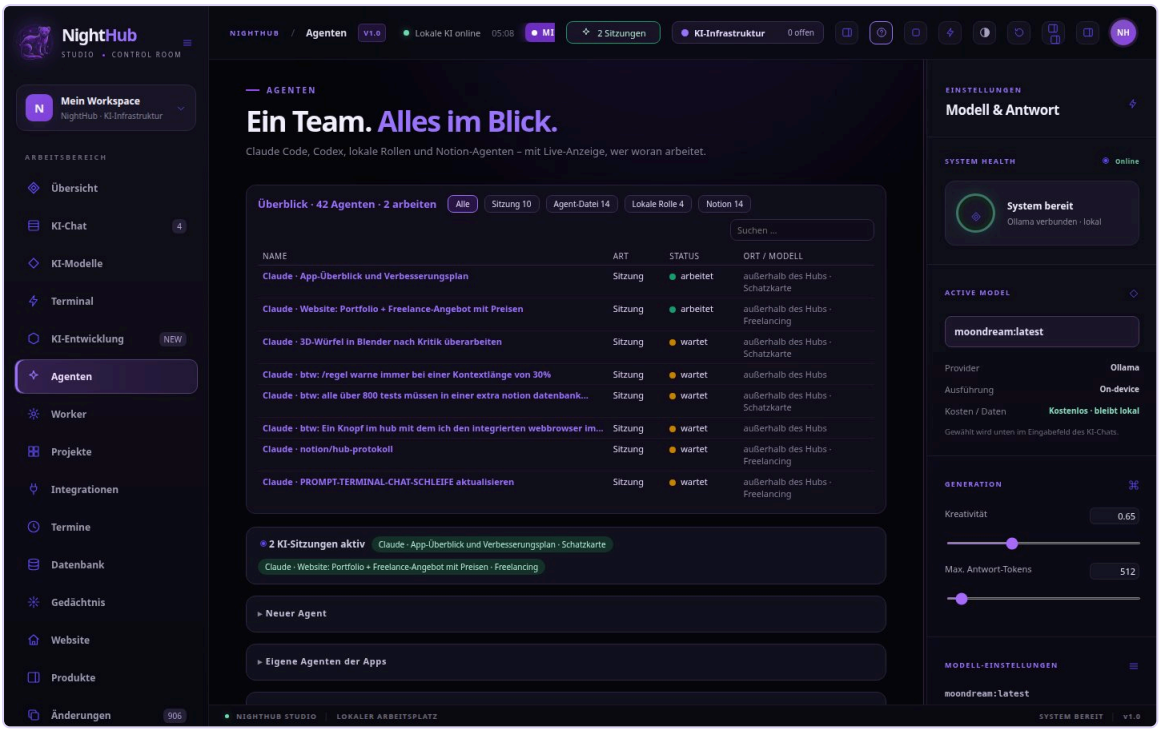
All providers in one place: status, enter keys (stored only in a protected file, never shown in the interface), choose the model and test with one click. Below: local models with speed and memory display.

04 AI development (NightLab)



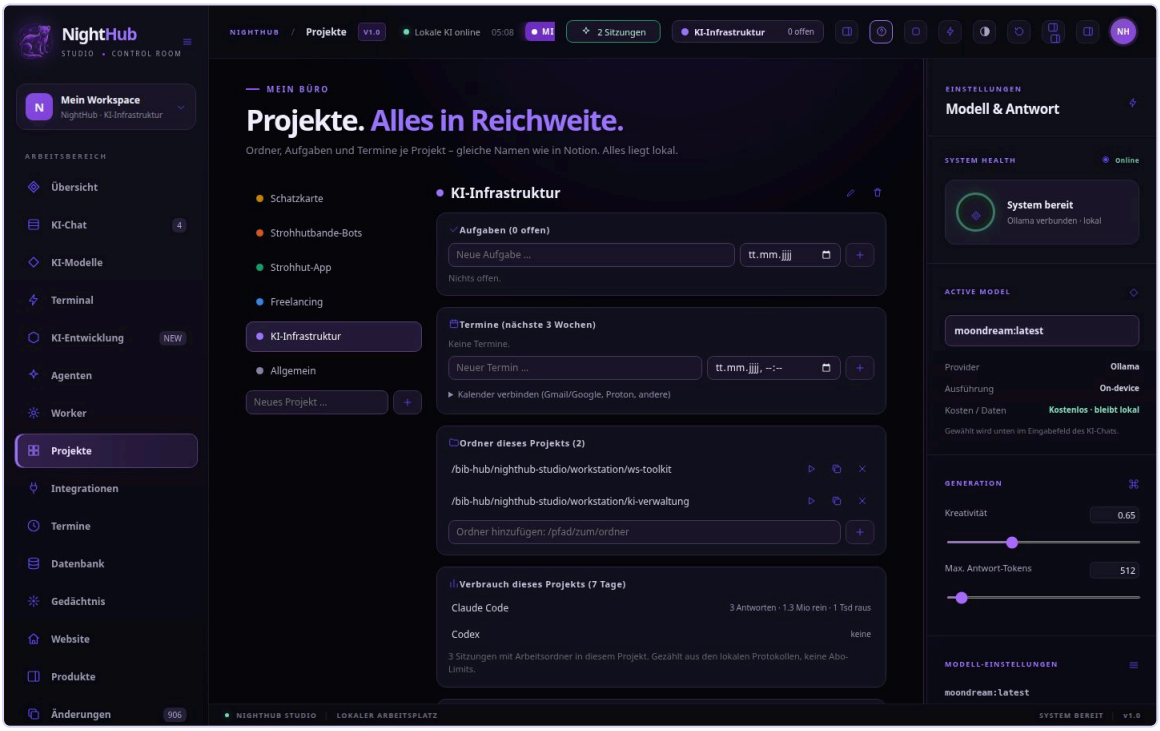
Two local models plan and work in turns on a shared task: plan, start, evaluate, archive. In sandbox mode they control figures in a 3D world (Godot), with no write access outside the experiment.

05 Agents



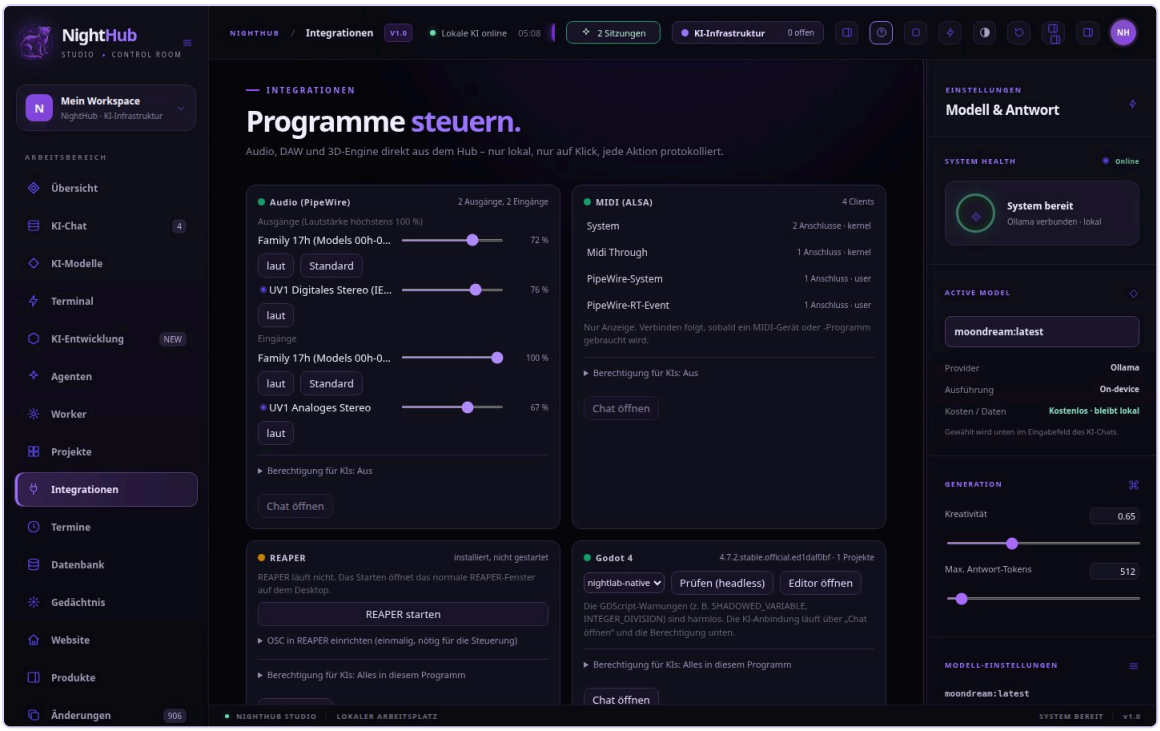
Live overview of all AI agents: Claude Code sessions, Codex, local roles and Notion agents – who is working, who is waiting, in which folder. New agents can be created and started here.

06 Projects



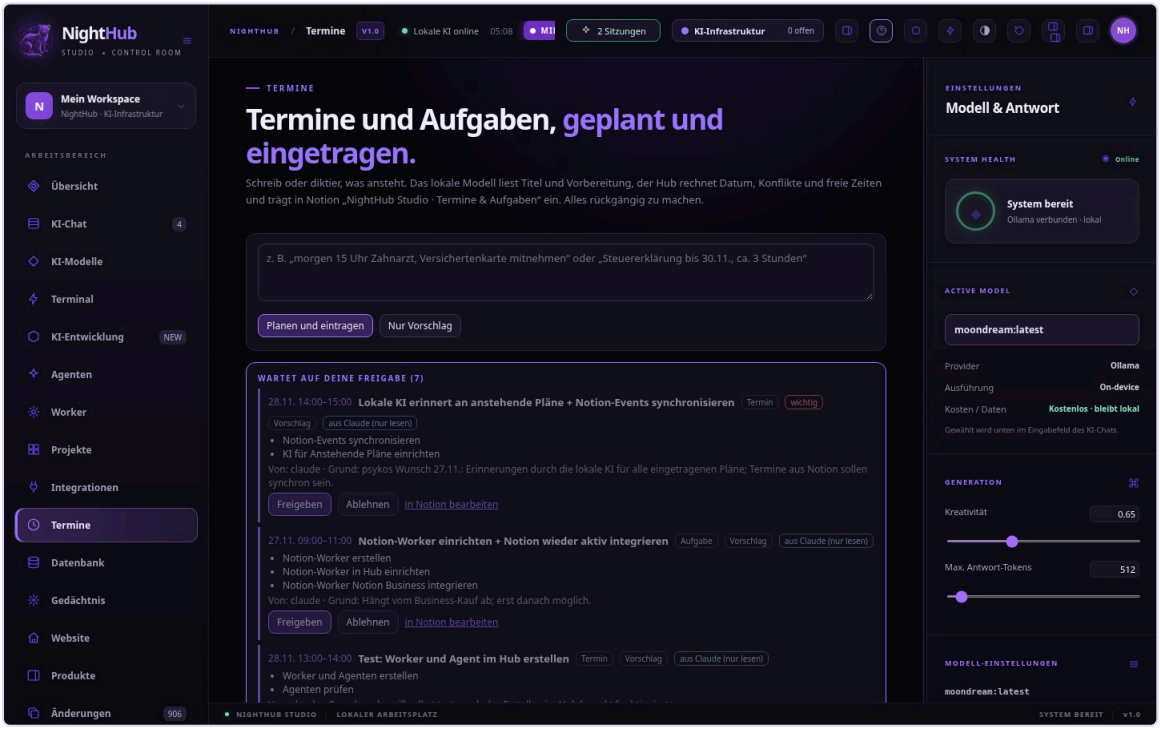
A small office: per project tasks, appointments, linked folders and AI usage of the last seven days. Same names as in Notion; everything is stored locally.

07 Integrations



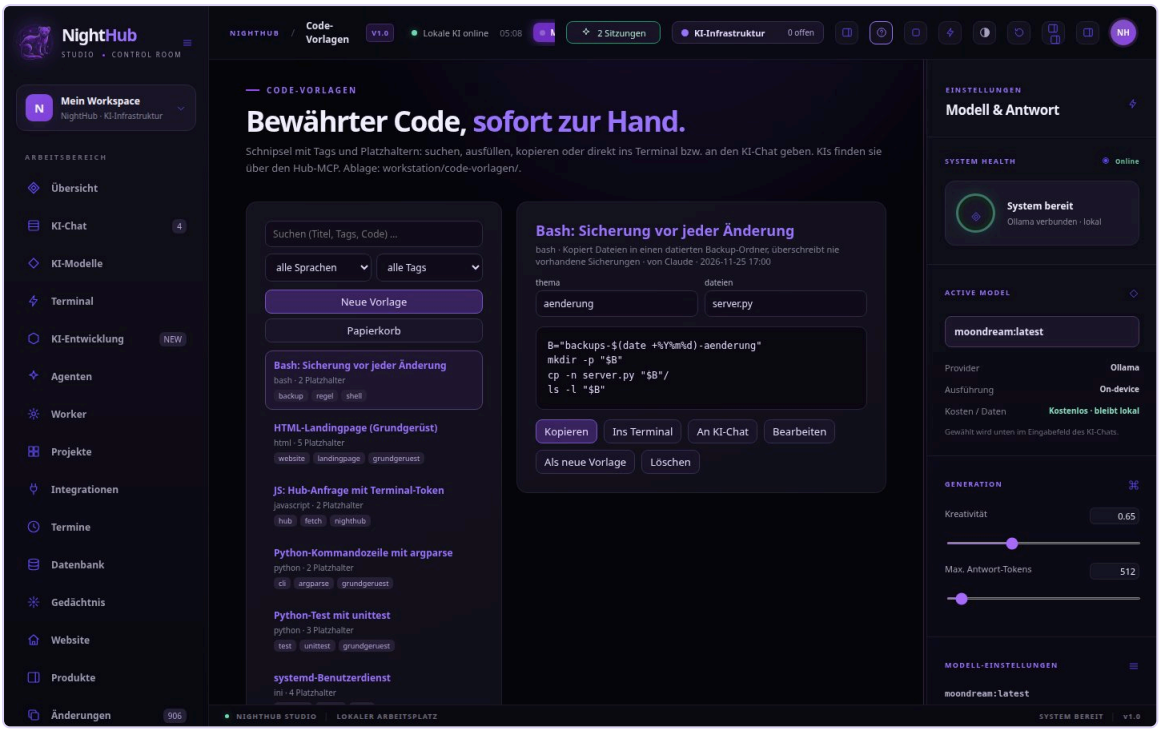
Control programs from the hub: audio (PipeWire), MIDI, REAPER, Godot and more. Each AI gets permissions per program; every action is logged. Plus a dedicated chat that may use the programs' tools.

08 Appointments



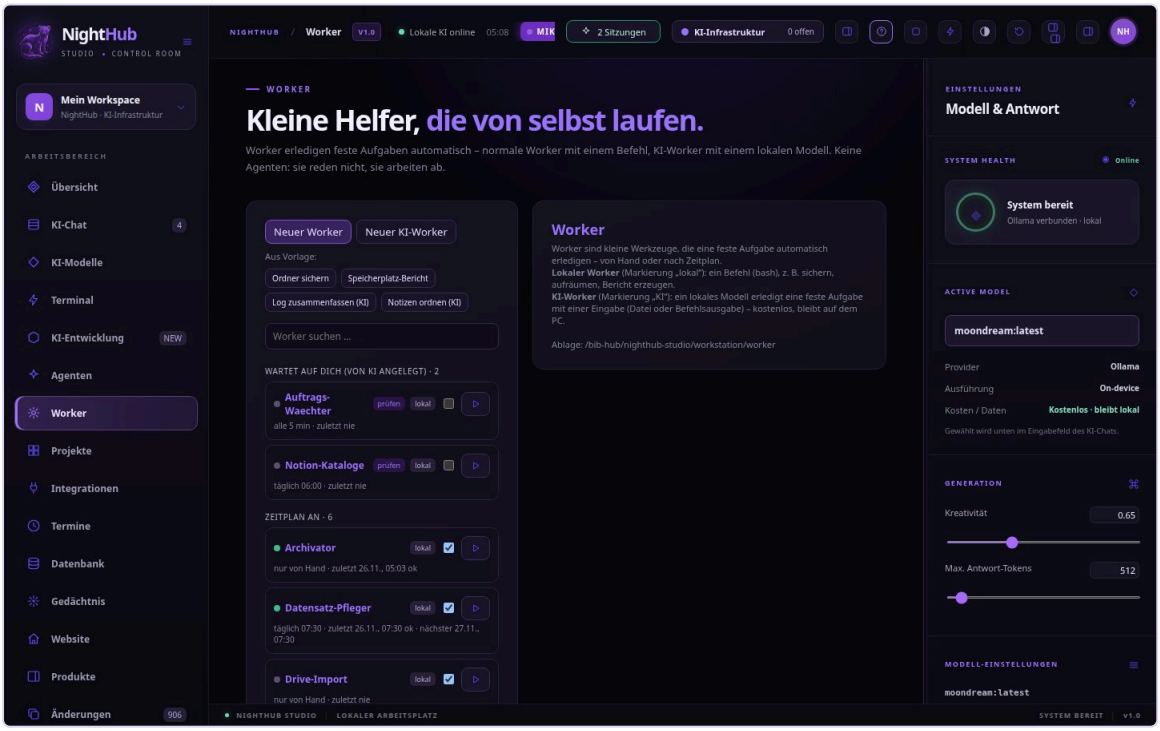
Write or dictate appointments in plain language. A local model extracts title and preparation; the hub calculates date, conflicts and free slots. Suggestions from all AIs wait for approval before they reach Notion.

09 Code snippets



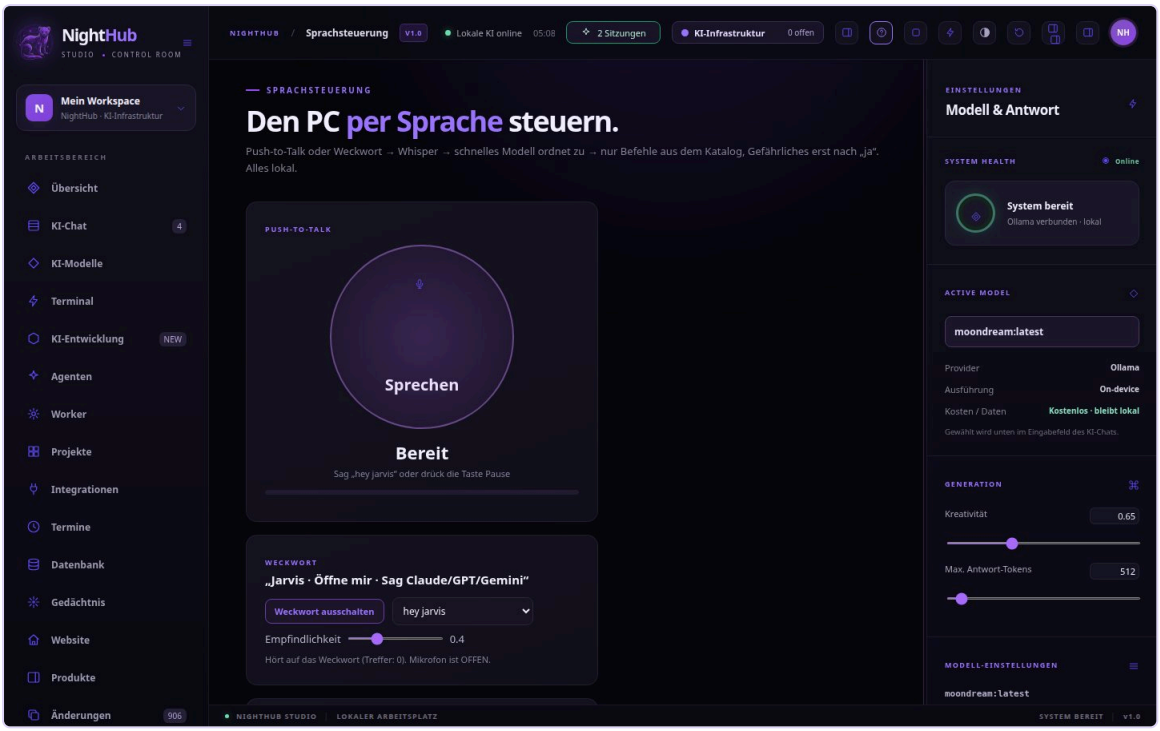
Proven code snippets with placeholders and tags: search, fill in, copy or send straight to the terminal or AI chat. Other AIs find and save snippets through the hub's MCP server.

10 Workers



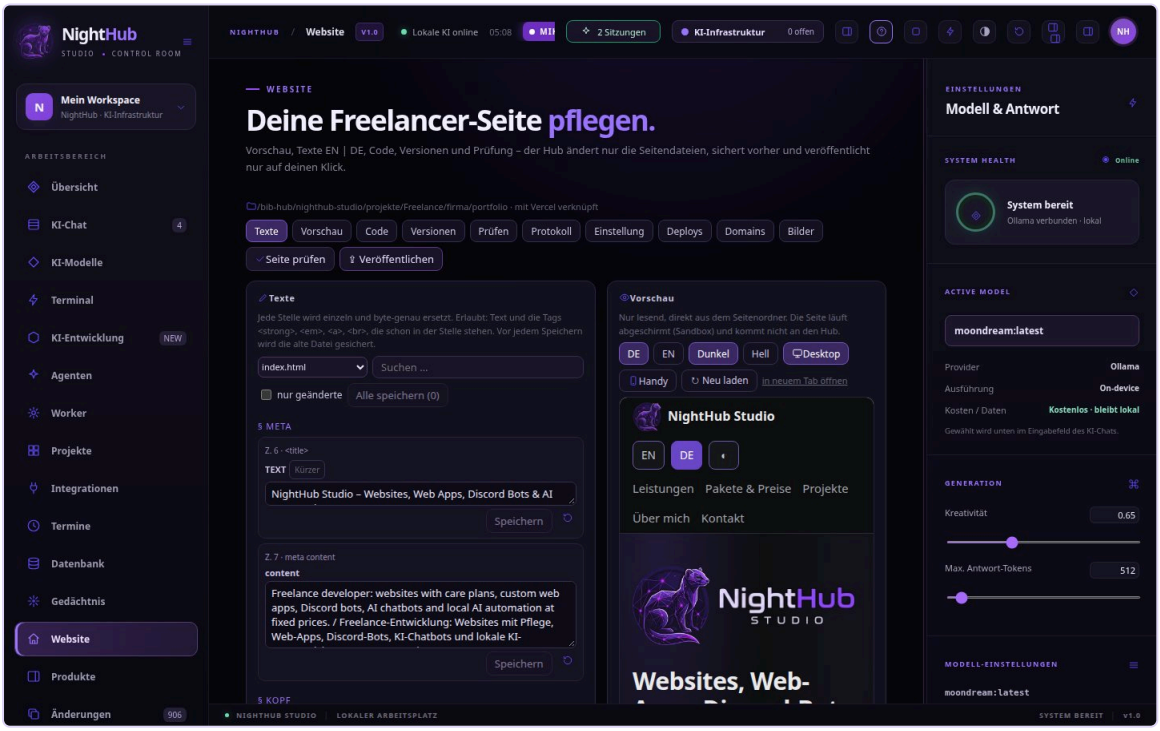
Small helpers for fixed tasks: local (one command) or with a local model. Schedule, log, output files, stop with one click. Workers created by an AI stay switched off until approved.

11 Voice control



Control the PC by voice: push-to-talk or wake word, Whisper on the GPU, a fast model maps the command – only commands from a fixed catalogue, anything risky only after a spoken "yes". All local.

12 Website care



Maintain the freelance website: edit German/English texts one by one, preview light/dark and mobile, code, versions, checks and publishing. The hub only changes the page files, backs up first and publishes only on click.